ArcGIS Server 9.3, FloatingPanels, and Javascript Frameworks

Contributed by Steve Gourley
17, Jul. 2008
Last Updated 22, Jul. 2008

When playing with AGS 9.3, firebug for Firefox , and fiddler for IE I realized that the esri:FloatingPanel control carries along a lot of baggage with it. Not only are the esri:FloatingPanels strictly asynchronous, every time you hide/show a panel with javascript, in our case 26KB of the contents of your FloatingPanel are sent from the client to server to maintain session state. Depending how much content you have in your FloatingPanel the overhead could make your application unusable for slower internet users.

Firebug and fiddler allow you to see the packets being sent back and forth between client and server and much much more. If you are a web developer, these 2 programs are a must.

But this brings me to the javascript frameworks. There are many javascript frameworks out there, just to name a few:

- Mootools
- Prototype
- Dojo
- YUI   These frameworks are broswer independent javascript libraries that allow you to do client side effects and rendering without have to worry about all the caveats of CSS and javascript browser rendering differences.

This was the route we chose to take to replace the esri:FloatingPanel. It may be a bit more javascript programming for us, but it will be a more lightweight solution for our clients in the long run.

We chose to use mootools at first based off of some examples of it's functionality that we were specifically interested in. Upon importing the mootools.js library into an ESRI web mapping application we saw one disappointing side effect: In internet explorer 6 and 7, we noticed a string of "undefined"'s being printed across the screen.

I'm no javascript guru and I still don't know the specific reason for this clash between ESRI and javascript so we switched to the prototype framework - Same problem.

Through deduction, I found that the ESRI controls containing esri:ContextMenus, eg: TaskResultsContainer, cause this problem. Like I said I'm not javascript guru so I don't completely understand what the issue is. My hunch is that these controls are implementing the same function name or fighting over an event of some sort.

I have submitted a ticket with ESRI support and shall report back my findings. Hopefully ESRI will play nicely with these frameworks in the future.

Updated 2008-07-21ESRI has confirmed that this is a bug (NIM036931).
"At this time, we are not aware of any work around." Updated 2008-07-22

From Rex Hanson,

This is actually a compatability issue with the Prototype library. The problem lies in how we're iterating over dense arrays in certain instances.   In general, there are two types of javascript for loop statements:

for (property in object)

and

for (index = 0; index < array.length; index++)

The first type of statement is most commonly used to iterate the properties of a javascript object (i.e. sparse array). It can also be used to iterate dense arrays (i.e. arrays declared explicitly as a new Array, with sequential numeric indexes starting at 0 and a length property). We have some cases in the ADF javascript where we've done just that - used the first type of statement to iterate dense arrays. For context menus this technique is used in the esriRenderContextMenus function. This introduces compatibility issues with Prototype because that framework adds properties to the Array object. So while the loop previously was fine because the array's "properties" were just array indexes, that's not the case with Prototype in the picture; inside the loop, object[property] always returns an element of the array without Prototype, but with Prototype it sometimes returns one of the extended Array's properties. Of course, the extended properties aren't of the type expected by the code, hence the error. In general the problem is usually solved by using the second type of for loop statement to iterate dense arrays.   Unfortunately IE always returns 0 for the Prototype Array.length property associated with the context menu array.

Updated 2008-07-22 #2

After a little more research, it appears our context menu javascript code does not utilize an Array object correctly... with or without prototype. The best option is to create an object array instead of an Array to store context menus.   You can fix it in your application by adding the following script block at the top of the form in your page. This will override the esriContextMenu variable created in the ADF's display_contextmenu.js.

```
<body>
   <form id="form1" runat="server">
   <script type="text/javascript" language="javascript">
      var esriContextMenus={};
   </script>
```

Note, this will work with or without a reference to the Prototype library. The fix will be included with ADF JavaScript in sp1.

Cheers,

-Rex